VRIJE UNIVERSITEIT AMSTERDAM

VU Research Portal

The Knowledge Graph for End-to-End Learning on Heterogeneous Knowledge

Wilcke, W.X.; Bloem, P.; de Boer, Viktor

2018

document version
Publisher's PDF, also known as Version of record

document license
Article 25fa Dutch Copyright Act

Link to publication in VU Research Portal

citation for published version (APA)
Wilcke, W. X., Bloem, P., & de Boer, V. (2018). *The Knowledge Graph for End-to-End Learning on Heterogeneous Knowledge*. Abstract from ICT Open 2018, Amersfoort, Netherlands.

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

• Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
• You may not further distribute the material or use it for any profit-making activity or commercial gain
• You may freely distribute the URL identifying the publication in the public portal ?

Take down policy
If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

E-mail address:
vuresearchportal.ub@vu.nl

Download date: 05. May. 2023

The Knowledge Graph for End-to-End Learning on Heterogeneous Knowledge*

Xander Wilcke, Peter Bloem, and Victor de Boer
{w.x.wilcke, p.bloem, v.de.boer}@vu.nl

Vrije Universiteit Amsterdam
Amsterdam, The Netherlands

Keywords: Knowledge Graph, Heterogeneous Knowledge, End-to-End Learning, Multimodal Embeddings, Semantic Web

Introduction

In modern machine learning, manual feature engineering has given way to end-to-end learning. Using end-to-end models, rather than selecting features by hand, data scientists can now simply feed the model the dataset as a whole, containing all relevant and irrelevant information, and trust the model to sift out relevant features automatically. However, most end-to-end models are domain specific and tailored to the task at hand, and therefore unsuited for learning on *heterogeneous knowledge*. In this paper, we argue that expressing this kind of knowledge using *knowledge graphs* enables us to build true end-to-end models across domains and use cases.

Knowledge graphs are build upon the principle that knowledge should be encoded using binary statements, called triples. Each of these triples relates exactly two resources (the *subject* and the *object*) via a relation (the *predicate*). We can more intuitively represent this as a directed graph, in which case each triple corresponds to an edge between two vertices. An example of such a graph is shown in Fig. 1, and tells us, amongst other things, that Pete is the brother of Kate, that Mary likes Pete, and that Mary is 32 years old. In the latter case, rather than linking two entities (things), we link an entity to a numerical literal, creating an attribute. Of course, literals can also be used to express other types of attributes, such as texts, dates, (hex-encoded) images, or other self-defined types.

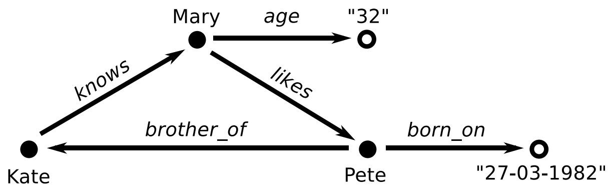

Figure 1: A small knowledge graph. Edges represent binary relations. Vertices' shapes reflect their roles: solid circles represent entities, opaque circles represent their attributes (literals).

Because knowledge graphs express all knowledge using the same simple encoding they are an attractive choice for learning on heterogeneous knowledge. After all, knowledge from one knowledge graph is encoded no differently than that from any other knowledge graph, irrespective of task and domain. If we develop machine learning models that can learn directly from this encoding, we can apply these models to any arbitrary knowledge graph, without first having to choose what features might be relevant to the learning task—making ad-hoc decisions and adding, removing, and reshaping information in the process. With feature engineering now being part of the model itself, it becomes possible to learn end-to-end.

Why is end-to-end learning so important? Solving a complex problem begins with breaking the problem up into subproblems, each one solved in a separate module. Any pre-processing done on the data, any manual feature extraction, harmonization and/or scaling can be seen as a module in the pipeline that cannot be tweaked, and does not allow a final optimization end-to-end. To solve this, we need to integrate these steps into the model itself, enabling an error signal to propagate though all of its modules, from the output back to the original data that inspired it. This has already been done successfully in various domains: to images, to sound, and to language. However, when

*This abstract is based on past [3] and current research.

faced with heterogeneous knowledge we often find ourselves resorting back to manual feature engineering. To avoid this, we need to adopt a data model capable of expressing heterogeneous knowledge naturally in various domains, in as usable a form as possible, and satisfying as many use cases as possible.

By adopting the knowledge graph as the data model for learning on heterogeneous knowledge we allow for true end-to-end learning across domains and use cases. The shift to knowledge graphs also provides other benefits to data scientists by a) greatly simplifying the integration and harmonization of datasets (we only need one shared entity/vertex) and by b) providing a natural way to integrate different forms of background knowledge (all knowledge uses the same encoding).

This idea suggests many research challenges. These include coping with incomplete knowledge, (how to fill the gaps), implicit knowledge (how to exploit implied information), and differently-modelled knowledge (how to deal with topological diversity). Yet another challenge is how to process multimodal knowledge—information of different types—which is the main focus of our current research.

Multimodal Graph Learning

Multimodal learning on knowledge graphs has been left largely unaddressed [3]. Instead, most present methods solely learn from graphs' structure: literals are either omitted completely or are stripped from their values and treated as non-literals. In either case we lose potentially relevant information which could have otherwise been exploited by our learning methods. To avoid this, we must treat literals and non-literals as separate cases. We must also address each data type separately and accordingly: text as strings, numbers as ordinal values, identifiers as nominal values, et cetera.

We can accomplish this by projecting the different modalities into the same representation space [1]. A large part of this work can already be done by the convolutions and pooling layers of a deep neural network, which merge the input signals into a lower-dimensional joint representation. However, to create meaningful multimodal embeddings we need to optimize on the latent variables, rather than on the most relevant unimodal feature (maximizing) or on a grey blend of features (averaging). This ensures that our embeddings stay close to the entities they represent.

To achieve this, we are looking into extending the Relational Graph Convolutional Network (RGCN) [2], which learns from graphs' structure in an end-to-end fashion, to additional modalities (Fig. 2). Special attention is given to spatial information (e.g. coordinates), which is an intrinsic aspect of all physical entities, and which enables us to perform spatially-oriented learning tasks.

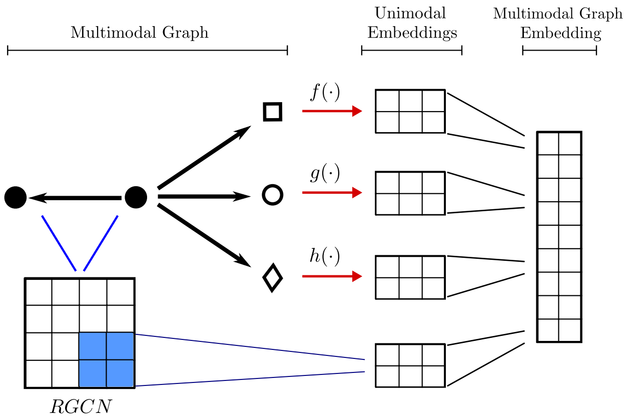

Figure 2: Schematic architecture for multimodal learning on knowledge graphs. The graph's structure is exploited by the RGCN, whereas the other modalities (depicted as various opaque shapes) are addressed using dedicated modules. Unimodal signals are then merged in a joint space to form multimodal graph embeddings.

Different from most other research on multimodal learning we are not trying to map different modalities of the same "thing" into a joint representation (e.g. an image and its description). Rather, we believe that by including as much information as possible, staying as close as we can to the original and complete knowledge, enables our methods to create a better internal representation of the entities we are trying to learn over, and therefore increases the overall performance of these methods.

References

[1] Tadas Baltrušaitis, Chaitanya Ahuja, and Louis-Philippe Morency. Multimodal machine learning: A survey and taxonomy. *arXiv preprint arXiv:1705.09406*, 2017.

[2] Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. Modeling relational data with graph convolutional networks. *arXiv preprint arXiv:1703.06103*, 2017.

[3] Xander Wilcke, Peter Bloem, and Victor de Boer. The knowledge graph as the default data model for learning on heterogeneous knowledge. *Data Science*, 1(1-2):39–57, 2017. DOI:10.3233/DS-170007.